

**A608 : PomoStamp**

삼성SW청년아카데미 서울캠퍼스 7기

공통 프로젝트 : 2022-07-11 ~ 2022-08-19

**포팅 매뉴얼**

담당 컨설턴트 : 박찬국

이도연(팀장), 이선호, 김현지, 박현영, 최진우

**<목차>**

**1. 기술 스택**

**2. 빌드 상세내용**

**3. 배포 특이사항**

**4. DB 계정**

**5. 프로퍼티 정의**

**6. 외부 서비스**

**7. Git ignore 파일**

모여서 공부하는 모각공이 유행하는 요즘에 시간 관리, 일정 관리가 추가로 제공되었으면 좋겠다고 생각하지 않으셨나요? 저희 PomoStamp는 모여서 공부하는 공간 제공과 함께 TodoList로 일정 관리, Pomo로 시간 관리를 제공합니다. 추가로 여러분의 집중력을 올리기 위해 각 공부 시간마다 집중력을 분석하여 제공해 본인의 공부 습관을 되돌아 볼 수 있고 최적의 공부 시간인 pomo를 추천해드립니다.

통합 서비스인 저희 PomoStamp 여러분도 Pomos와 함께 공부해 보세요☺

**1. 기술 스택**

가. 이슈 관리 : jira

나. 형상 관리 : Gitlab

다. 커뮤니케이션 : MatterMost, Notion

라. 개발 환경

1) OS : WINDOW 10

2) IDE

가) IntelliJ 22.1.3

나) Visual Studio Code 1.70.0

3) DataBase

가) MySQL Workbench 8.0.21

나) Redis 0.9.3.817

4) Server : AWS EC2

가) Ubuntu 20.04 LTS

나) Xshell 7.0.0025

다) FileZilla 3.60.2

라) Nginx

마. 상세 사유

1) Backend

가) Gradle 7.5

나) Flask 서버 2.1.3

다) Java open-JDK zulu 8.33.0.1

2) Frontend

가) Nodejs LTS 16.16.0

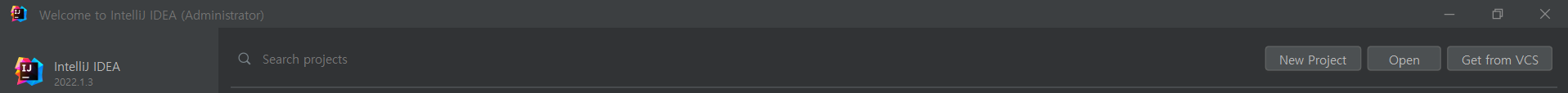
나) React(CRA) 18.2.0

다) OpenVidu 2.22.0

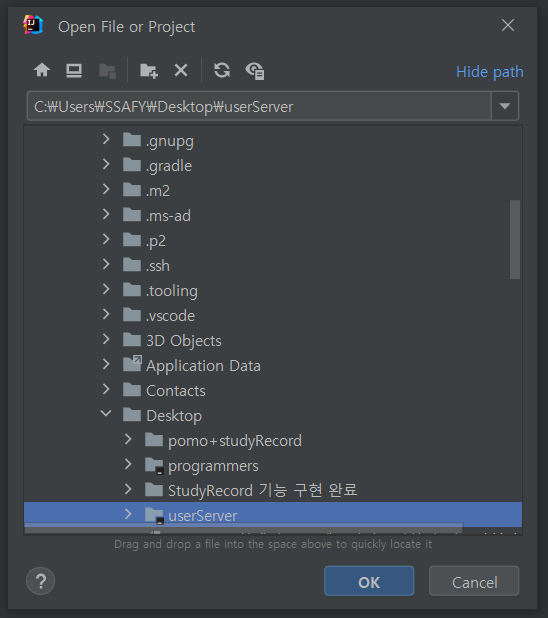
**2. 빌드 상세내용**

가) PomoStamp – backend

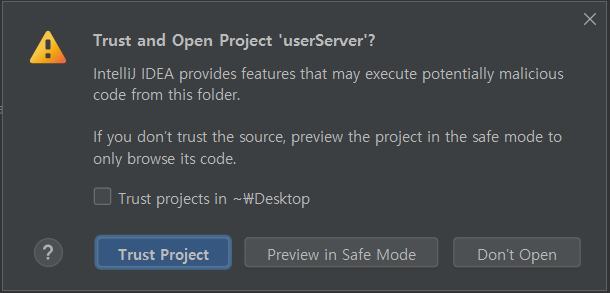
가 - 1) PomoStamp Intellij 에서 실행하는 방법



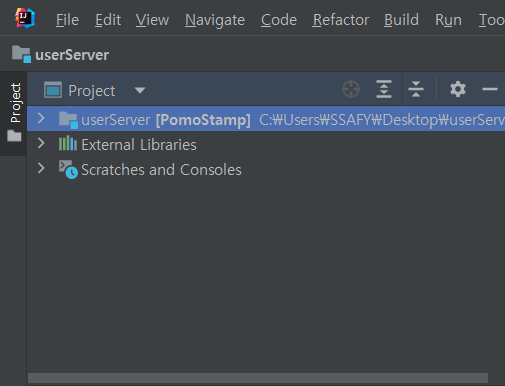
1. 인텔리제이 실행 후 Open 클릭합니다.

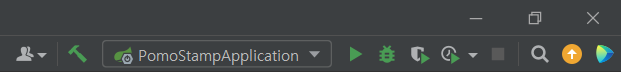


1. userServer 선택 후 OK 버튼 클릭합니다.



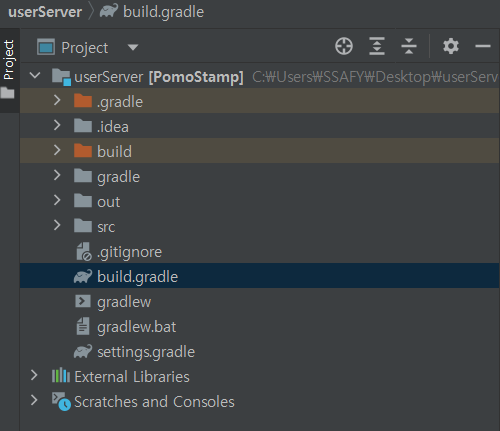
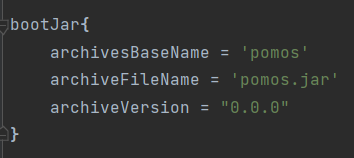
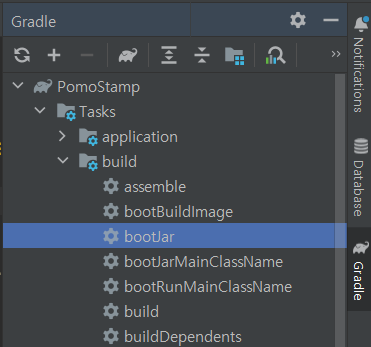
1. Trust Project 클릭합니다.



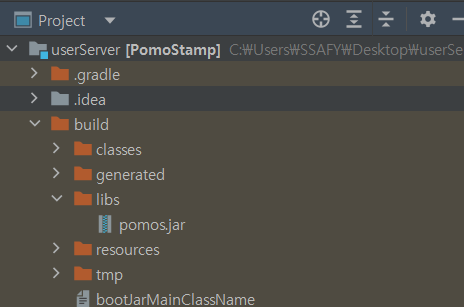


1. Project에 userServer 가 import 되어 있는거 확인 후 초록색 재생버튼 클릭하면 application이 실행됩니다.

가 - 2) PomoStamp jar 파일 생성하는 방법

1. build.gradle 파일 선택 -> archivesBaseName 과 archiveFileName의 .jar 앞에 동일한 이름으로 원하는 jar 파일 명 작성 -> Gradle – build – bootJar 더블 클릭합니다.



1. build – libs 폴더 안에 pomos.jar 파일이 생성된 걸 확인할 수 있습니다.

나) PomoStamp – frontend

1. 패키지 관리를 위한 yarn 설치

npm i –g yarn

1. node\_modules를 위한 기본 install

yarn install

1. 빌드하기

yarn build

**3. 배포 특이사항**

가. AWS EC2에 만들어진 jar 파일 4개와 dist 폴더를 업로드 후, 자동배포를 진행합니다.

나. 다음과 같은 명령어로 확인합니다.

1) 현재 구동중인 jar 파일 확인

ps -ef | grep “pomos” / **ps** -ef | grep java

해당 명령어를 통해서 현재 실행중인 jar 파일 프로세스의 PID를 볼 수 있습니다

2) 구동중인 프로세스 종료

Kill -9 <PID>

해당 명령어를 통해서 프로세스를 종료하고, 구동중인 파일을 확인한 후, 종료가 정상적으로 된 것을 확인하면 수정된 파일을 배포합니다.

3) 새롭게 백그라운드에서 서버 실행

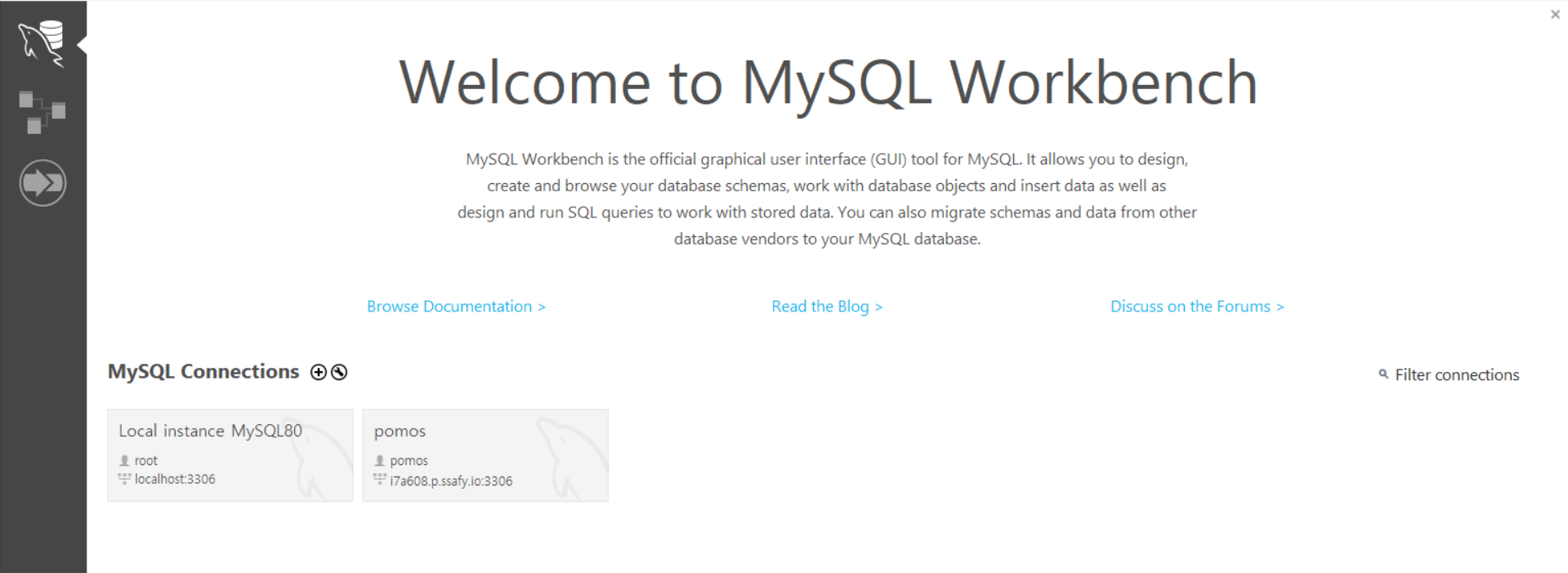
nohup java –jar {파일이름}.jar &

해당 명령어를 통해서 백그라운드에서 서버가 계속 실행되도록 합니다.

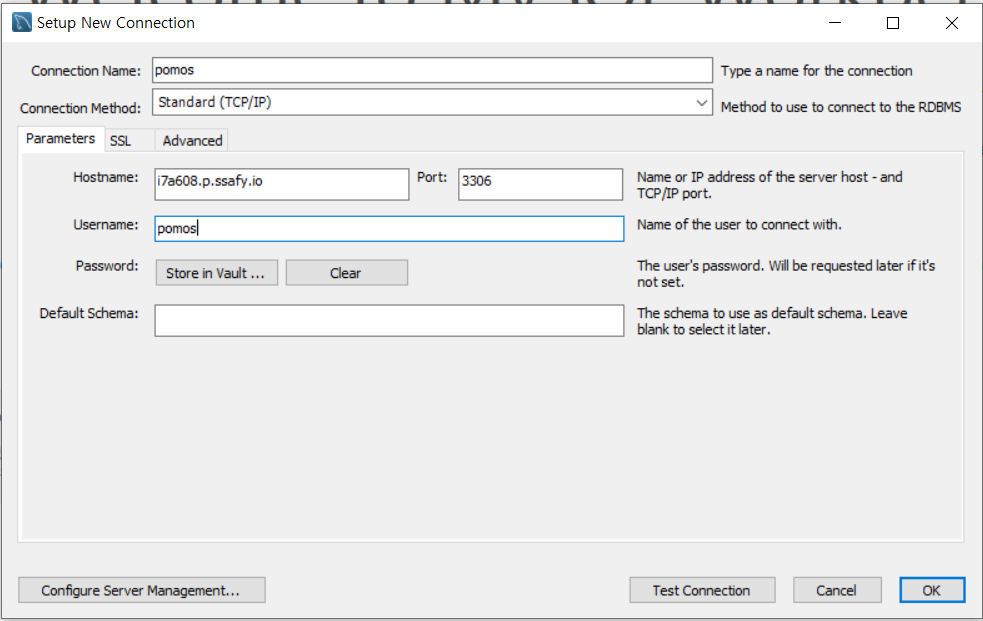
다. 배포가 정상적으로 진행된 후 서버에서 확인하면서 마무리합니다.

**4. DB 계정**

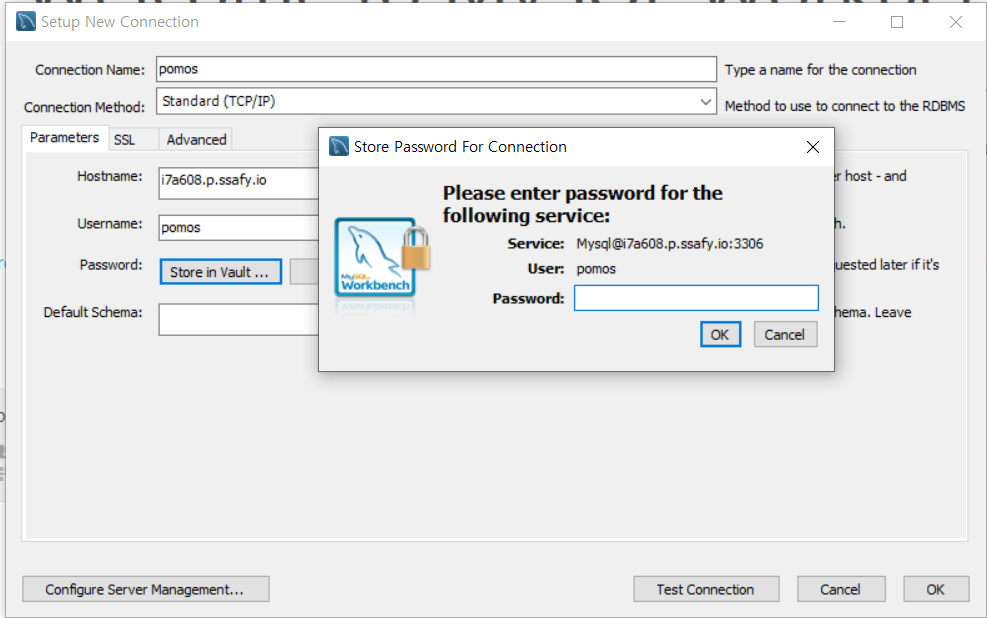
가) MySQL Workbench 추가하기



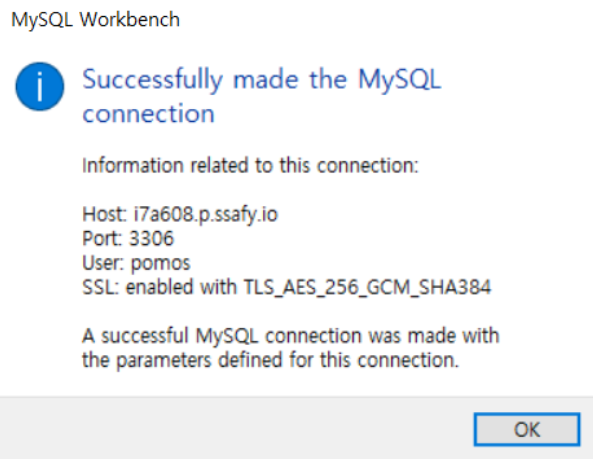
1. MySQL Workbench 를 열어서 새로운 계정을 추가하기 위해 + 버튼을 클릭합니다.



1. Hostname : i7a608.p.ssafy.io , Port : 3306 , Username : pomos 입력하고 store in Vault 클릭합니다.



1. Password : pomos1229 입력 후 OK 클릭 -> Test Connection 클릭합니다.



1. 위와 같은 모달창이 나오면 Connection 성공 입니다.

**5. 프로퍼티 정의**

**가) Docker**

가 – 1) 사전 확인

1. 리눅스 커널이 최소 3.10버전 이상이어야 합니다.
2. 22년 2월 기준으로 확인결과, 2021년 4월 30일부터 우분투 16.04 LTS는 더 이상의 도커 릴리즈 지원이 종료되었습니다.
3. 기존의 오래된 도커 버전이 있다면 삭제하고 설치해주세요

sudo apt-get remove docker docker-engine docker.io containerd runc

가 – 2) Docker 설치

sudo apt-get update # package index 업데이트

# HTTPS를 통해 repository를 이용하기 위한 package 설치

sudo apt-get install

ca-certificates

curl

gnupg

lsb-release

# 도커의 official GPG Key를 등록

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

# stable repository 등록

echo

"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \ $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# 도커 엔진 설치

sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io

docker --version # 20.10.17, build 100c701(로컬과 동일, 최신 버전)

**나) NGINX**

나 – 1) NGINX설정

1) ubuntu에 nginx 설치

sudo apt install nginx

2) nginx 설정 파일 변경

sudo vi /etc/nginx/sites-available/default

3) nginx 시작

sudo service nginx start

4) nginx 중지

sudo service nginx stop

nginx의 설정을 하기 위해서 멈춘 후, 설정하고 재시작합니다

나 - 2) Front에 NGINX 빌드하기

1) node 설치하기

curl -sL https://deb.nodesource.com/setup\_16.x | sudo -E bash –

sudo apt install nodejs

2) 패키지 관리를 위해 yarn 사용

npm i -g yarn

3) 빌드가 완료된 후, 폴더 이동하여 module 설치

cd ./front/pomostamp

yarn install

모듈 설치해줍니다

yarn build

이 명령어를 통해서 빌드가 되면, i7a608.p.ssafy.io로 접속합니다

**다. openvidu(화면공유)를 위한 SSL 인증서 발급**

다 – 1) SSL 인증서 발급 및 적용

1) 관리자 계정으로 변환

sudo su

2) opt 폴더로 이동

cd /opt

3) openvidu 설치

curl https://s3f-eu-west-1.amazonaws.com/aws.openvidu.io/install\_openvidu\_lastest.sh

4) openvidu로 이동

cd openvidu

5) DOMAIN\_OR\_PUBLIC\_IP 과 OPENVIDU\_SECRET을 넣어주기

nano .env

이후, DOMAIN\_OR\_PUBLIC\_IP=i7a608.p.ssafy.io

OPENVIDU\_SECRET=MY\_SECRET

CERTIFICATE\_TYPE=letsencrypt

LETSENCRYPT\_EMAIL=i7a608.p.ssafy.io

로 처리해줍니다

6) openvidu 실행

./openvidu start

다 - 2 ) . NGINX 에 openvidu 실행을 위한 SSL 인증서 저장 및 분기 설정

1) 설정 파일로 이동

cd /etc/nginx/sites-enabled

2) default 파일 실행

nano default

3) SSL 인증서를 적용한 default파일 구성

server {

listen 443 ssl default\_server;

listen [::]:443 ssl default\_server;

ssl\_certificate /etc/letsencrypt/live/i7a608.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/i7a608.p.ssafy.io/privkey.pem;

server\_name i7a608.p.ssafy.io;

location / {

root /home/ubuntu/front/front/pomostamp/build;

index index.html index.htm index.nginx-debian.html;

try\_files $uri $uri/ /index.html;

location /pomo/ {

proxy\_pass http://localhost:8080;

proxy\_http\_version 1.1;

proxy\_set\_header Connection "";

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Forwarded-Host $host;

proxy\_set\_header X-Forwarded-Port $server\_port;

}

}

server {

listen 8081;

listen [::]:8081;

server\_name i7a608.p.ssafy.io;

return 301 https://i7a608.p.ssafy.io$request-uri;

}

-openvidu에서 여러가지 포트를 사용하고 있기 때문에 443포트를 사용해서 설정했습니다

-location을 통해서 /는 프론트, /pomo는 백엔드로 분기처리해서 필요한 요청을 처리했습니다,

다 – 3 ) NGINX 재시작

sudo service nginx restart

**라. AWS EC2 DB 세팅**

라 – 1) 접속 정보

* IP주소/Host명: i7a608.p.ssafy.io
* root 계정 PW: ssafy1229
* user 계정
  + ID: pomos
  + PW: pomos1229

라 – 2) 초기 설정 명령어

라 – 3) 유저 관리

# EC2 접속

ssh -i I7A608T.pem [ubuntu@i7a608.p.ssafy.io](mailto:ubuntu@i7a608.p.ssafy.io)

sudo ufw allow ssh

sudo ufw enable

# 초기 설정

sudo apt-get update # 패키지 매니저 업데이트

sudo apt-get install mysql-server # mysql 설치

sudo ufw allow mysql # 외부 접속 기능 설정(포트 3306 오픈)

sudo systemctl start mysql # mysql 실행

sudo systemctl enable mysql # mysql 서버 재시작 시 mysql 자동 재시작

# mysql 접속

sudo /usr/bin/mysql -u root -p # 비밀번호: ssafy-> ssafy1229로 변경! # mysql > [여기에 명령어 입력]

1. Mysql 계정 조회

show databases;

use mysql;

// mysql 5.x 는 "password" 또는 "authentication\_string"으로 확인

mysql> select Host, User, Password from user;

mysql> select Host, User, authentication\_string from user;

// mysql 8.x

mysql> select Host, User, authentication\_string from user;

1. Root 계정 비밀번호 변경 (자동 난수 생성)

// mysql 5.x

mysql> update user set password=password('1234') where user='root';

mysql> update user set authentication\_string=password('1234') where user='root';

// mysql 8.x

mysql> alter user 'root'@'localhost' identified with mysql\_native\_password by '123';

1. 적용 후 종료

mysql> FLUSH PRIVILEGES;

mysql> exit

Bye

1. 외부 접속 허용을 위해 mysqld.cnf 파일 수정

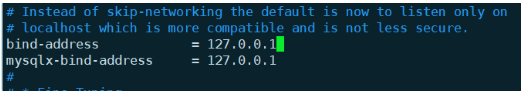
sudo su # root 권한 부여

cd /etc/mysql/mysql.conf.d # mysqld.cnf 파일이 있는 디렉터리로 이동

vi mysqld.cnf # 파일 실행

# bind-address 부분을 찾아 0.0.0.0으로 변경

service mysql restart # mysql 서버 재실행



//외부 접속을 위한 user 생성, 권한 부여

//%는 모든 곳에서 허용한다는 의미, username/password 원하는 것으로

mysql> create user 'username'@'%' identified by 'password';

mysql> grant all privileges on \*.\* to username@'%';

//user를 생성하지 않고 root계정을 원격에서 사용하려면 host를 변경

mysql> update user set Host='%' where User='root';

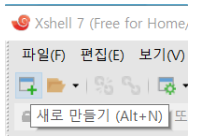
**마) XShell**

마 – 1 ) Xshell 설치 방법

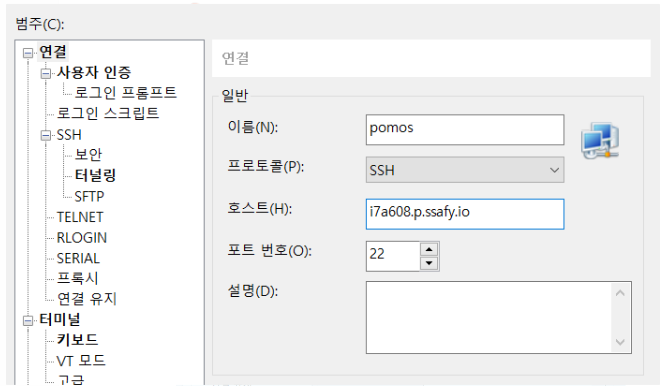
https://www.netsarang.com/en/free-for-home-school/

1. 위의 사이트에서 이름과 이메일을 입력하면 이메일로 다운로드 링크가 옵니다.
2. 링크를 검색창에 복사 붙여넣기 하면 자동으로 Xshell 프로그램이 다운로드 됩니다.

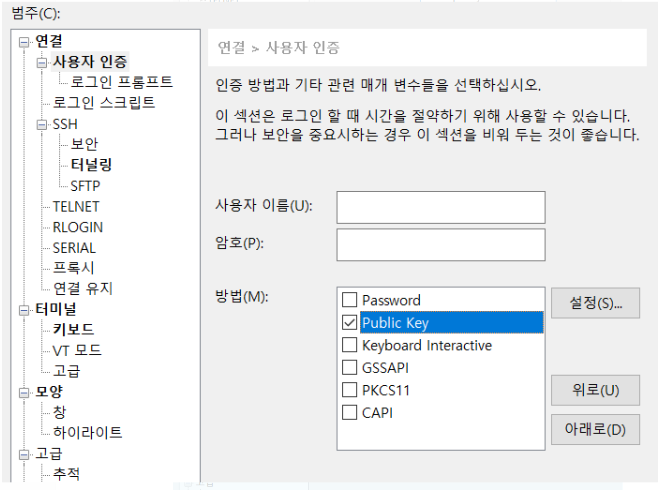
마 – 2) Xshell 초기 설정 방법



1. 새로 만들기 아이콘을 클릭합니다.

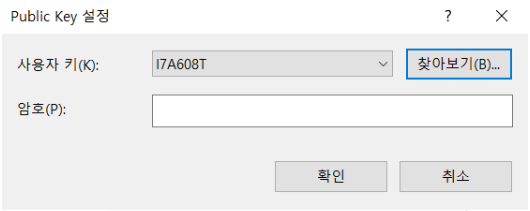


1. 원하는 이름과 호스트를 설정합니다.

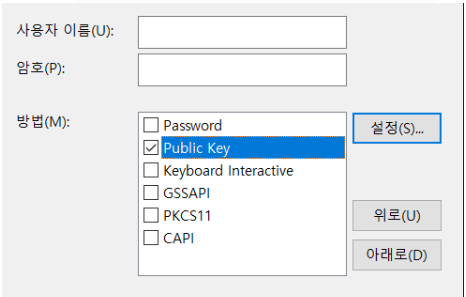


1. 왼쪽 메뉴에서 사용자 인증 → public key 체크합니다.

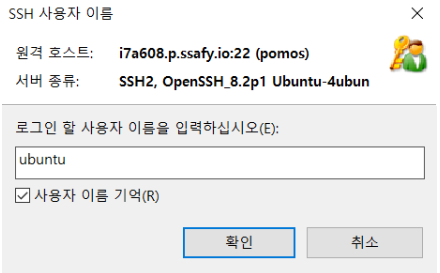
파랗게 보이도록 클릭해야 오른쪽 설정 버튼 활성화되고 설정 버튼을 클릭해주세요



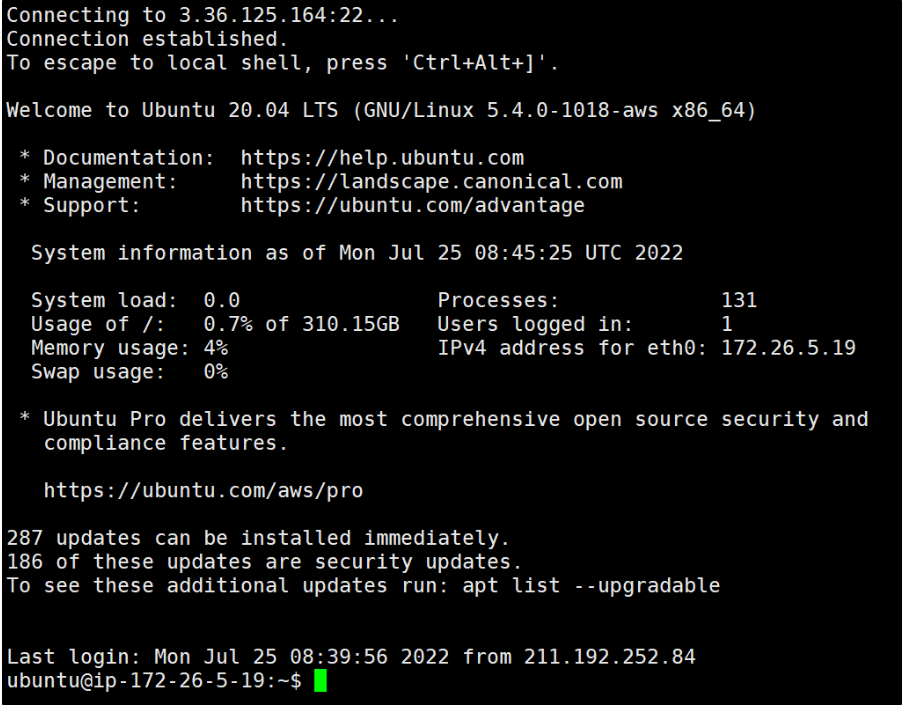
1. 찾아보기 → pem 파일 선택 → 암호 비워둔 채로 확인버튼을 클릭해주세요.



1. 사용자 이름과 암호를 비워둔 채로 연결 버튼을 클릭합니다.



1. 이름에 **ubuntu** 입력하고 사용자 이름 기억 체크합니다.



1. 위와 같은 화면이 뜨면 서버 접속 성공입니다.

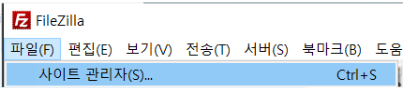
**바) FileZila**

바 – 1) FileZila 설치

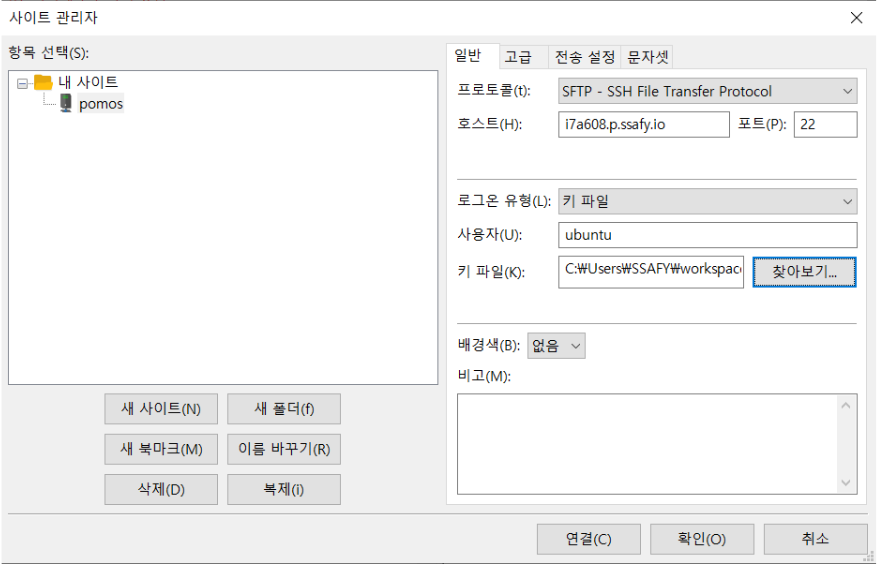
https://filezilla.softonic.kr/download

1. 위의 사이트에서 FileZila 프로그램을 다운받아주세요.

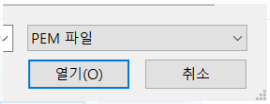
바 – 2) FileZila 초기 설정



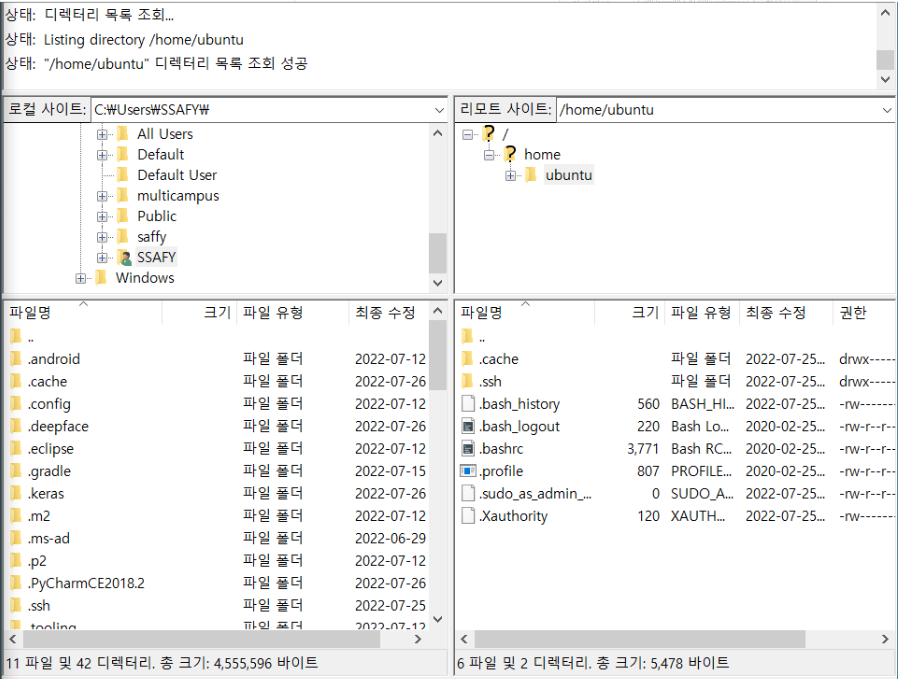
1. 파일 -> 사이트 관리자를 클릭합니다.



1. 새 사이트(이름 자율) -> 프로토콜 SFTP 선택 -> 호스트 입력 -> 포트 22 입력 -> 로그온 유형(키 파일) 선택 -> 사용자(ubuntu) 입력 -> 키 파일 찾아보기 버튼을 클릭합니다.



1. 확장자를 PEM 파일로 설정해야 PEM 파일이 보이고 그 파일을 선택해주세요.



1. 위와 같은 화면이 나오면 연결 성공입니다.

**사) Redis 서버**

사 – 1) Redis 서버 설치 및 관리

# 우분투에 redis 설치

wget https://download.redis.io/redis-stable.tar.gz

tar xzf redis-stable.tar.gz

cd redis-stable

sudo make install

# 빌드 완료 후 확인

cd src

# conf 파일 상관없이 실행

./redis-server

# 작성한 conf 파일을 기반으로 redis 실행

sudo ./redis-server /etc/redis/redis.con

# redis 상태 확인

systemctl status redis-server.service

# src 디렉토리에서 실행

./redis-cli

# 설정한 비밀번호를 입력해야 접근 가능

AUTH {password}

사 – 2) 설정 파일 관리

# 우분투 redis 설정파일 위치

sudo vi /etc/redis/redis.conf

#### redis.conf 기본 설정하기####

# redis 접속 허용 IP 설정

bind [IP] # ex: bind 127.0.0.1

# redis 허용 포트 설정

port [포트번호] # ex: port 6379

# 접속 패스워드 설정

requirepass [패스워드] # ex: requirepass foobared

1. 우분투로 Redis 설정 파일 관리하는 코드 입니다.
2. 외부 서버가 redis로 접속하기 위해 허용할 IP를 지정하는 부분이며 개별 IP, IP 대역으로 지정이 가능하며 모든 IP에서 허용할 경우 0.0.0.0 으로 작성하시면 됩니다.

**아) Flask 서버**

아 – 1) 초기 환경 설정

mkdir analysisServer

cd analysisServer

python3 -V # python만 치면 버전2 결과가 나옴 sudo apt-get install flask

아 – 2) 서버 접속 및 기본 코드 실행

cd analysisServer

python3 start\_app.py

# 정상적으로 실행이 되면 i7a608.p.ssafy.io:5000 로 접속

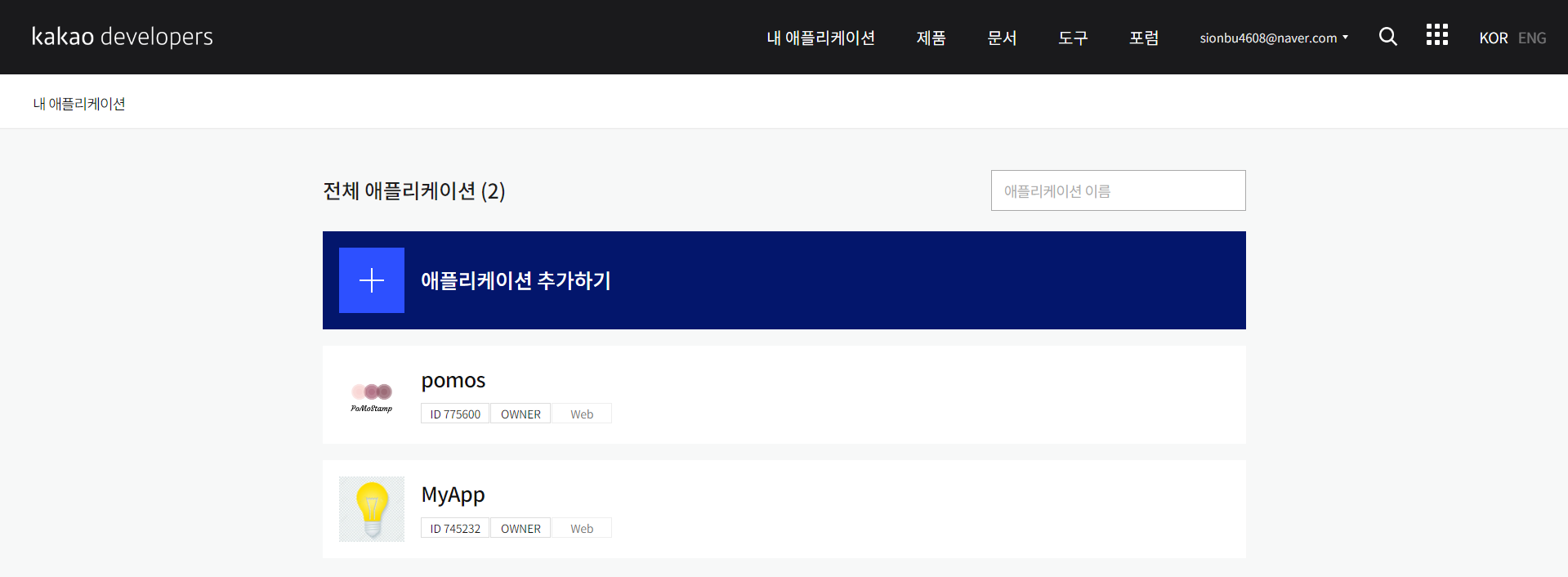
**6. 외부서비스**

가) 카카오 로그인

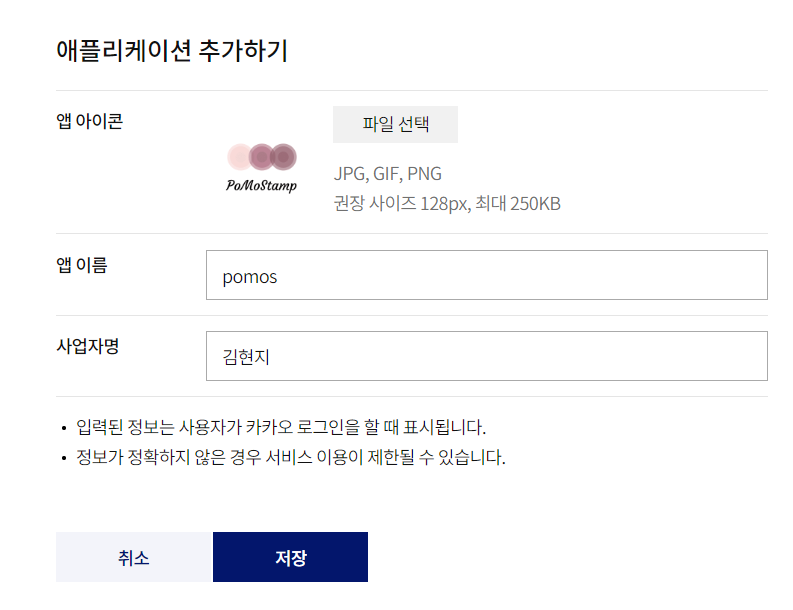
카카오 API를 이용하여 별도의 회원가입 절차가 생략되어 로그인 시 사용자의 편의성을 제공합니다.



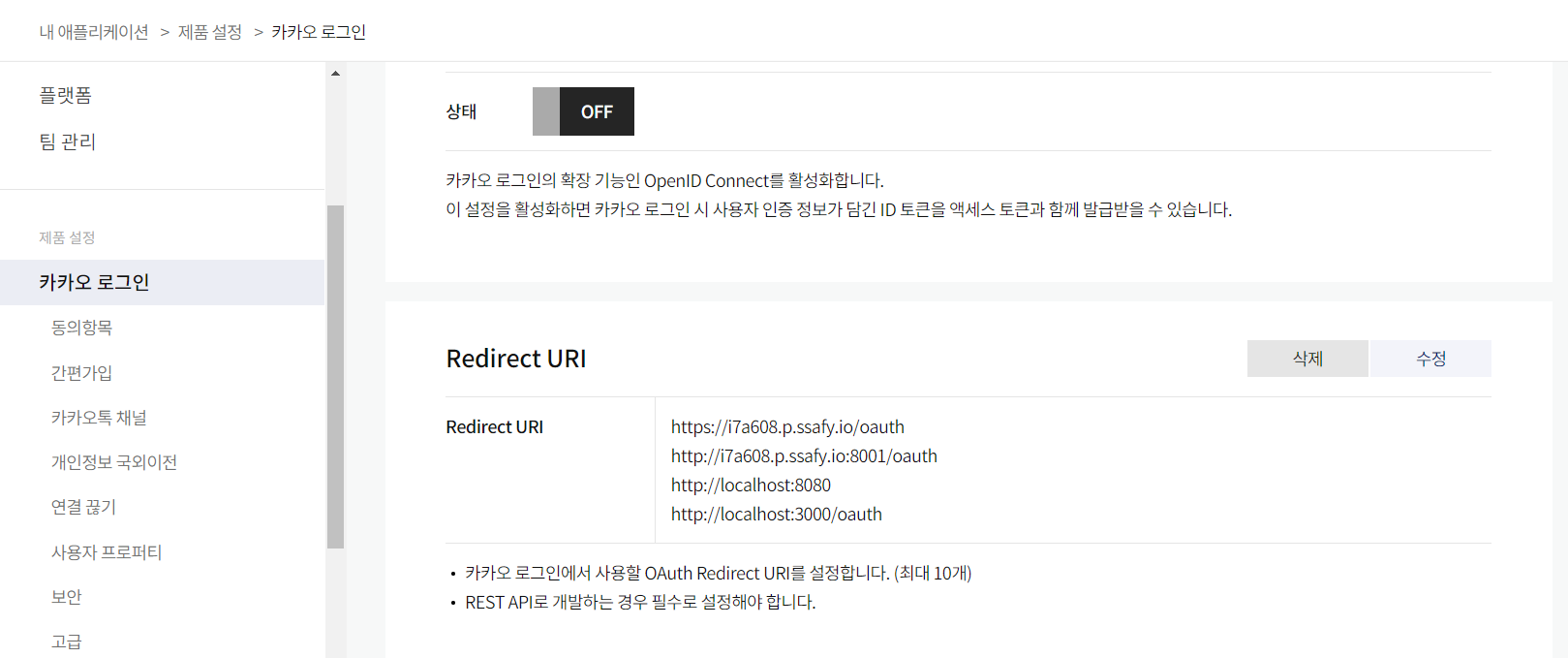
1. Kakao developers 사이트에 접속하여 로그인 후 내 어플리케이션을 클릭합니다.



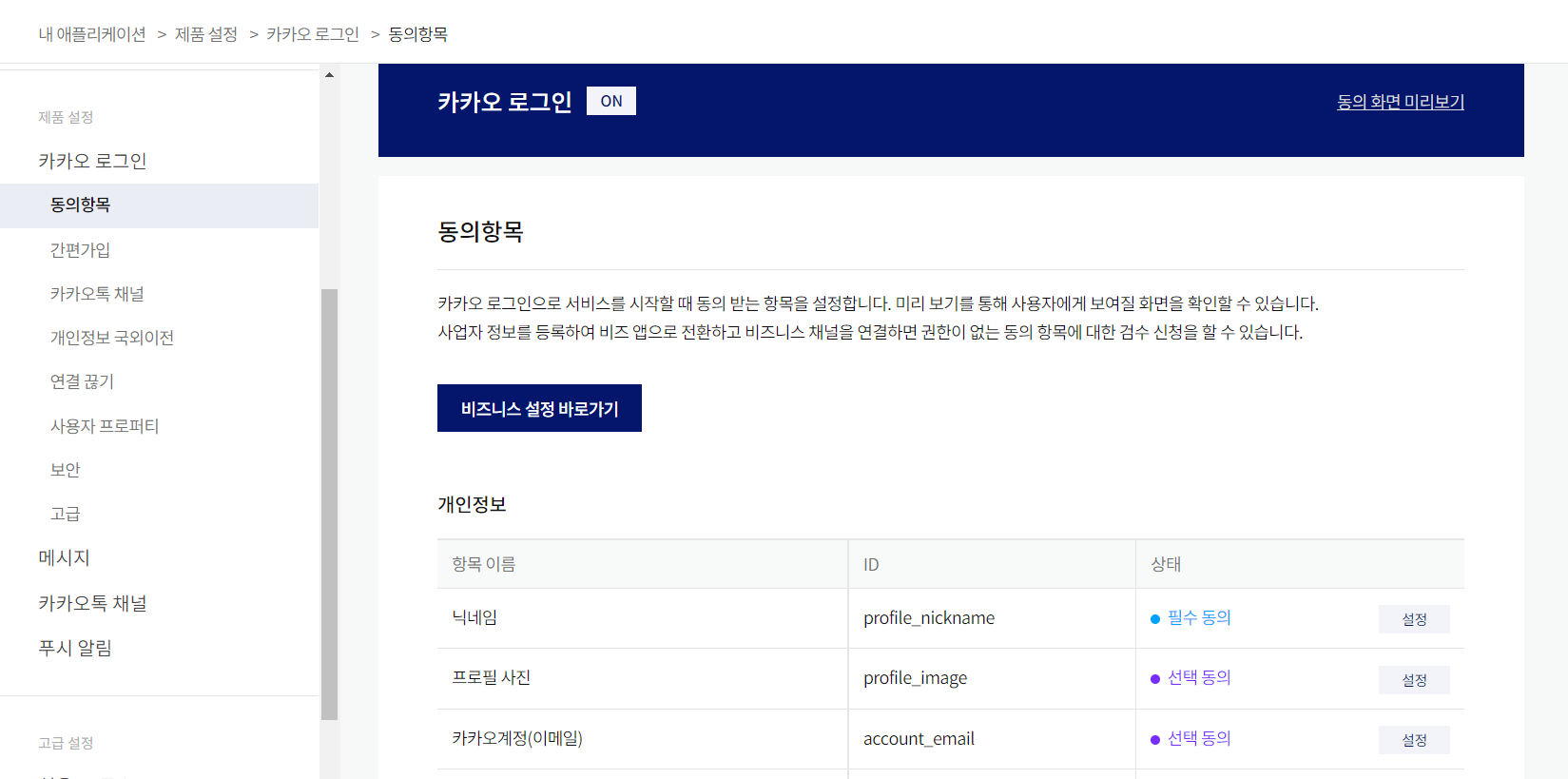
1. 어플리케이션 추가하기 버튼 클릭합니다.



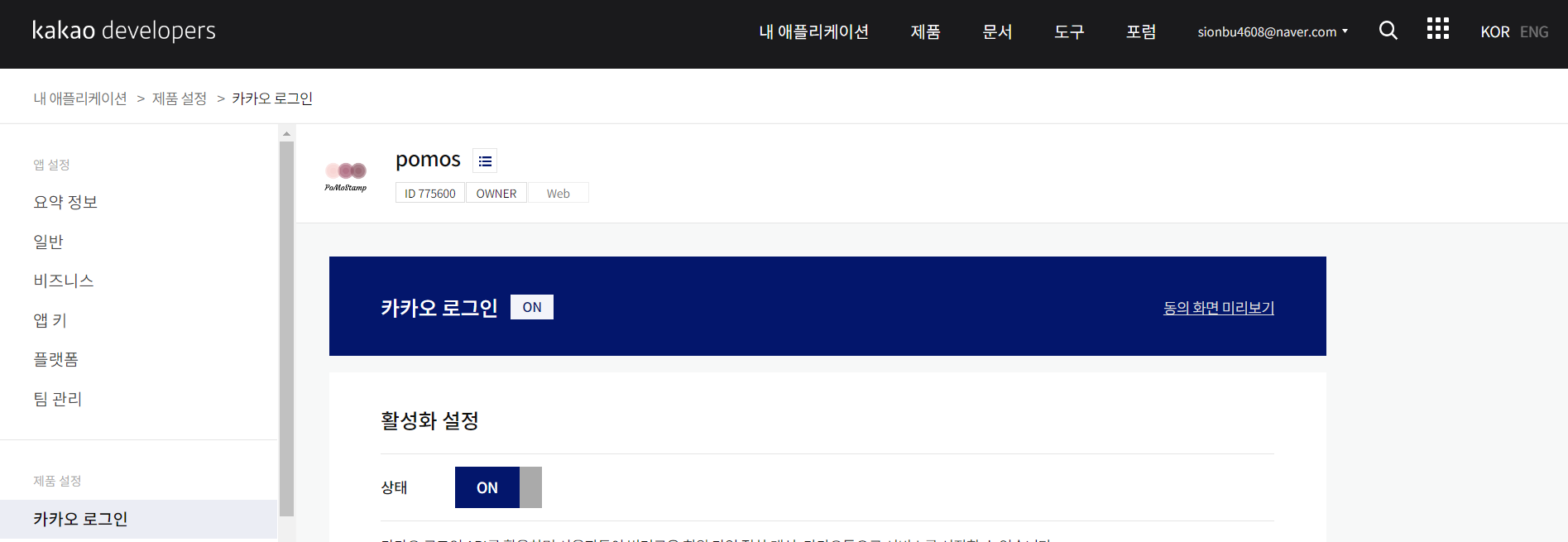
1. 앱 이름 : pomos , 사업자명 : 이름 , 앱 아이콘을 선택 후 저장버튼을 클릭하면 오른쪽 사진과 같이 앱이 생성된 걸 확인할 수 있습니다.



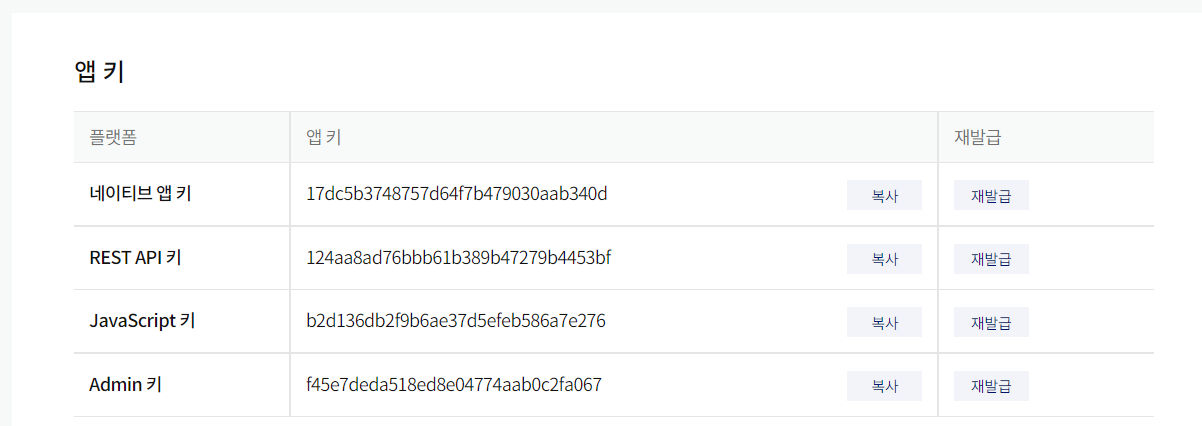
1. Redirect URL 설정



1. 사용자 동의 항목을 구성합니다.



1. 카카오 로그인 활성화 설정을 ON으로 클릭합니다.



GET

oauth/authorize?client\_id=${REST\_API\_KEY}&redirect\_uri=${REDIRECT\_URI}&response\_type=code

HTTP/1.1

Host: kauth.kakao.com

1. 카카오 로그인 인가 코드 받아오는 요청 URL

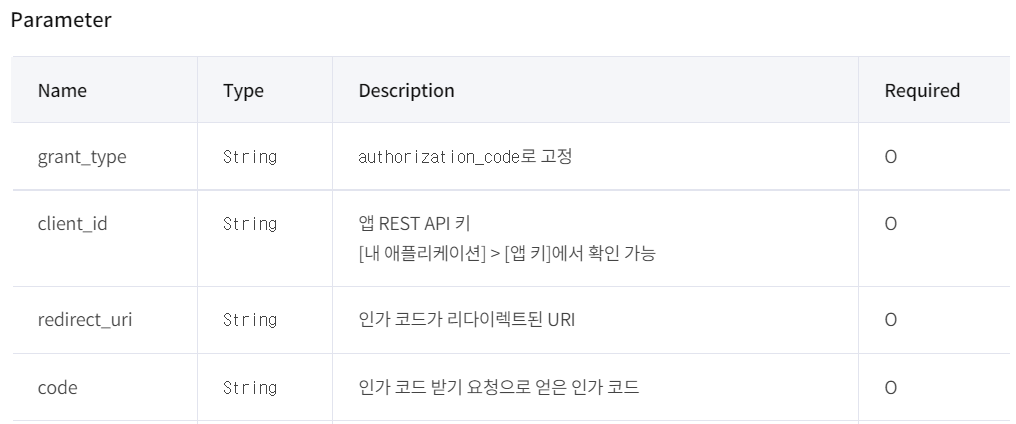
POST

/oauth/token

HTTP/1.1

Host: kauth.kakao.com

Content-type: application/x-www-form-urlencoded;charset=utf-8



1. 카카오 토큰 받는 요청 URL , Parameter

GET/POST

/v2/user/me HTTP/1.1

Host: kapi.kakao.com

Authorization: Bearer ${ACCESS\_TOKEN}

Content-type: application/x-www-form-urlencoded;charset=utf-8

1. 사용자 정보 요청하는 URL

**7. Git ignore 파일**

**가) .env 파일**

가 - 1) 작성 방법

REACT\_APP\_END\_POINT=http://localhost:8080

REACT\_APP\_KAKAO\_CLIENT\_ID=124aa8ad76bbb61b389b47279b4453bf

REACT\_APP\_LOGIN\_REDIRECT\_URI=http://localhost:3000/oauth

1. Local 에서 실행시키기 위한 코드로 userServer에서 CORS 처리를 위해 작성해놓은 URL과 카카오 로그인 redirect url을 https ://i7a608.p.ssafy.io→ http://localhost:3000로 경로 수정해야 합니다.

REACT\_APP\_END\_POINT=https://i7a608.p.ssafy.io

REACT\_APP\_KAKAO\_CLIENT\_ID=124aa8ad76bbb61b389b47279b4453bf

REACT\_APP\_LOGIN\_REDIRECT\_URI=https://i7a608.p.ssafy.io/oauth

1. AWS 에서 실행시키기 위한 코드입니다.

가 – 2) 파일 위치

1. Pomostamp 프로젝트 바로 하단 구조에 위의 내용이 적힌 .env 파일을 만들어 넣어줍니다.

(src 폴더 안으로 들어가지 말고 package.json 파일과 같은 위치로 넣어주세요.)